

09/892,684

MS174305.01 / MSFTP256US

**AMENDMENTS TO THE CLAIMS**

This listing of claims will replace all prior versions, and listings, of claims in the application.

**Listing of Claims:**

1. (Currently amended) A system that facilitates employment of a pluggable formatter, comprising:
  - a decision module that retrieves a first data structure as a graph of objects for serialization thereof;
  - a plurality of rule sets that define serialization information about data structure types;
  - a surrogate that specifies serialization instructions for all objects of a given type;
  - and
  - a serialization selector that determines a rule set of the plurality of rule sets to provide to the decision module based on the data structure type, the decision module populating a second data structure based on the serialization information and providing a pluggable formatter with the second data structure, so that the pluggable formatter can serialize the second data structure to an externalized format defined by the pluggable formatter, the pluggable formatter is selected from a variety of pluggable formatters employed on the system.
2. (Previously presented) The system of claim 1, the rule set is definable in the data structure.
3. (Previously presented) The system of claim 1, the rule set is defined in a third party file.
4. (Previously presented) The system of claim 1, the rule set is a default format based on markings in the first data structure.

09/892,684

MS174305.01 / MSFTP256US

5. (Currently amended) The system of claim 1, the first data structure being a graph of objects, ~~an object~~.
6. (Cancelled)
7. (Previously presented) The system of claim 1, further comprising an object ID generator coupled to the decision module, the object ID generator assigns object IDs to each object in the graph of objects.
8. (Original) The system of claim 1, the data structure containing information within the data structure that the serialization selector utilizes in determining a rule set.
9. (Previously presented) The system of claim 1, the decision module is integrated into the pluggable formatter.
10. (Cancelled)
11. (Currently amended) A system that facilitates employment of a pluggable formatter, comprising:
  - a surrogate file that implements serialization instructions for all objects of a given type;
  - a pluggable formatter selected from a variety of employed pluggable formatters that produces a decoded serialized stream based at least in part on the serialization instructions of the surrogate file;
  - a formatter services component that receives [[a]] the decoded serialized stream from [[a]] the pluggable formatter and creates a data structure as a graph of objects for deserialization of the decoded serialized stream; and
  - an object manager that tracks data in the decoded serialized stream and determines forward references to additional data to provide fixups to the data structure upon receipt of the additional data.

09/892,684

MS174305.01 / MSFTP256US

12. (Previously presented) The system of claim 11, further comprising a serialization selector that determines a rule set of a plurality of rule sets to provide to the formatter services component based on the data structure type being deserialized, the formatting services component populates the data structure based on the determined rule set.
13. (Previously presented) The system of claim 12, the rule set is definable in the data structure.
14. (Previously presented) The system of claim 12, the rule set is in a third party file.
15. (Cancelled)
16. (Previously presented) The system of claim 11, the formatter services component reinstantiates each object of a graph of objects and the object manager performs fixups on the objects after the graph of objects have been deserialized.
- 17-31. (Cancelled)
32. (Currently amended) A method for deserialization of a stream into a graph of objects, the method comprising:
- implementing a predefined rule set for all objects of a certain type via a surrogate;
  - selecting a pluggable formatter from a variety of employed pluggable formatters
  - to provide a decoded serialized stream;
  - receiving [[a]] the decoded serialized stream from [[a]] the pluggable formatter;
  - retrieving from the decoded serialized stream an object from a graph of objects;
  - determining an object type for deserialization;
  - instantiating an uninitialized instance of the object type;
  - selecting a serialization rule set from one of a user defined rule set in the object and a user defined rule set in another object and a default rule set; and

09/892,684

MS174305.01 / MSFTP256US

populating the uninitialized instance of the object type based on the selected serialization rule set.

33. (Original) The method of claim 32, further comprising registering the object with an object manager.

34. (Original) The method of claim 32, further comprising tracking forward references to other objects within the object.

35. (Original) The method of claim 32, further comprising repeating retrieving, determining, instantiating and populating for each object in an object graph.

36. (Original) The method of claim 35, further comprising performing fixups on the objects in the object graph caused by forward references.

37-48. (Cancelled)

49. (Currently amended) A system that facilitates making formatters pluggable, the system comprising:

means for providing serialization information of an object from a graph of objects to a pluggable formatter through assignment of a surrogate file, so that the pluggable formatter can serialize the object in a selectable externalized format, the pluggable formatter is chosen from a variety of pluggable formatters implemented on the system;

means for tracking deserialization of the object outside the pluggable formatter;

means for instantiating an uninitialized instance of an object type, so that the object information can populate the object type; and

means for performing fixups on the objects due to forward references.

50. (Cancelled)